

## Setup for PMAC Accessory 36

### Using PMAC's A/D Card for Data Acquisition and Servo Feedback Loop Control

#### Connecting Accessory 36 to PMAC with battery-backed CPUs

When connecting Acc-36 to a PMAC with battery-backed memory, connect the J1 connector to PMAC's JEXP connector and make sure jumper E11 is open for the first Acc-36. When connecting several Acc-36s to a PMAC with battery backed memory, connect the first Acc-36 (J1) to PMAC (JEXP) and then connect the next Acc-36 from J3 to the previous Acc-36 at its J2 connector. The table below summarizes the connection to PMAC with battery backed CPU.

From		To	Jumper E11
Acc-36 #1	J1	PMAC (JEXP)	Open
Acc-36 #2	J2	Acc-36 #1 J3	Closed
Acc-36 #3	J2	Acc-36 #2 J3	Closed

#### Connecting Acc-36 to PMACs with Flash CPUs

If connecting only one Acc-36 to a PMAC with flash CPU, then connect the 50-pin ribbon from the J2 connector to PMAC's JEXP connector and close the E11 jumper\*. When connecting several Acc-36s to a PMAC with Flash Memory, connect the first Acc-36 (J2) to PMAC (JEXP) and then connect the next Acc-36 from J3 to the previous Acc-36 at its J2 connector. The table below summarizes the connection to PMAC with flash CPU.

From		To	Jumper E11
Acc-36 #1	J2	PMAC (JEXP)	Closed*
Acc-36 #2	J2	Acc-36 #1 J3	Closed
Acc-36 #3	J2	Acc-36 #2 J3	Closed

Connecting to J2 instead of J1 will bypass the buffers on the accessory board since this PMAC option has on-board buffers. Connecting to PMAC in this manner makes the data transfer a much faster process.

#### Acc-36 Data Acquisition Test Setup

A simple test setup has been developed to help the first time user understand how this PMAC accessory works. A diagram of the system is shown on the following page. If a variable voltage power supply is not available, a potentiometer could be implemented easily in its absence. We are also going to wire a potentiometer to the output voltage of Acc-36 (pins 18 and 19) to simulate a sensor which draws its power from Acc-36. After Acc-36 has been properly connected to PMAC, a PLC will be written to monitor the voltage being read in the PMAC Executive watch window. The PLC that will be written for this exercise will be very similar to the example in the manual. In this case, the ADC1, ADC2, ADC3, ADC4, and ADC9 are being tested.

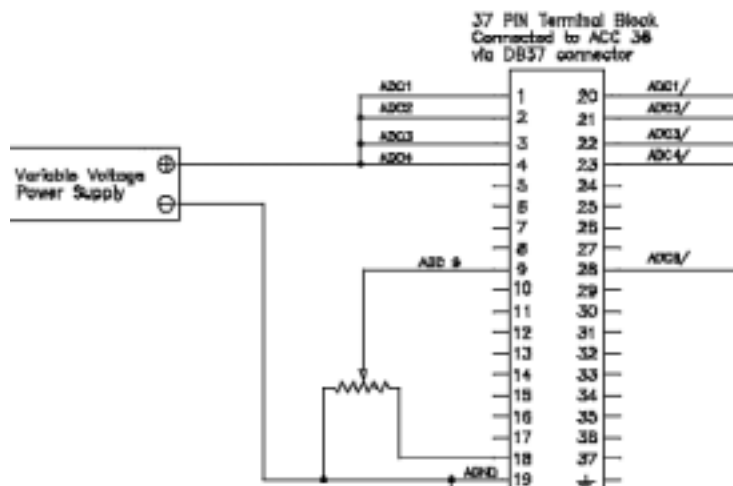
E1 and E7 are jumpered → \$FFD0 is the Y-memory address space used for Acc-36

Only one Acc-36 used  
Single ended unipolar inputs 16-1 → E11 is open  
channel Acc-36

The following M-Variable definitions and PLCs can be downloaded to PMAC and used for data acquisition.

Now, place P100, P101, P102, P103 and P108 into the watch window and then enable the PLC. Notice the value of the voltage is represented by a 12-bit number. If the inputs to Acc-36 are adjusted to 10V, the P-Variables should be reading 4095. Calibrate the system if a high degree of accuracy is needed for data acquisition purposes.

Test Setup for Accessory 36



Wiring Diagram

### Acc-36 Servo Feedback Test Setup

As mentioned in the Acc-36 manual, Acc-36's data may be used for servo loop feedback with PMAC using Firmware V1.15 or greater. The A/D card sends PMAC a 12-bit number to a user-defined address space of PMAC's processor bus. Variable I60 tells PMAC this address will be copied automatically into PMAC's memory at a high rate so that they can be used as servo feedback. PMAC will then place this information into the low 12-bits of registers Y:\$0708 to Y:\$070F (for ADC1 to ADC7) and X:\$0708 to X:\$070F (for ADC8 to ADC15). PMAC's Encoder conversion table then processes this information for servo loop control.

The following test setup will show how to use Acc-36 data for closed loop control. For this test, a potentiometer is used as a master for a motor which has encoder feedback to close its position and velocity loop. (An optional method to test this system would be to couple potentiometer to the end shaft of the motor and use the potentiometer to close the position and velocity feedback loops.) A diagram of the system is shown on the following page.

The encoder conversion table must be set up properly to read the data from Acc-36. As stated in the manual, the data must be read as parallel position data with filtering. For this example, the end of the default encoder conversion table (Y:\$72A) was selected as the entry address and since parallel position from Y word with filtering was chosen to process the data, the processed data address would be X:\$72C. The source address to process was the data for ADC1 was Y:\$0708. The encoder conversion table could be set up in the Executive Program dialog boxes or in the Executive Program terminal mode as follows:

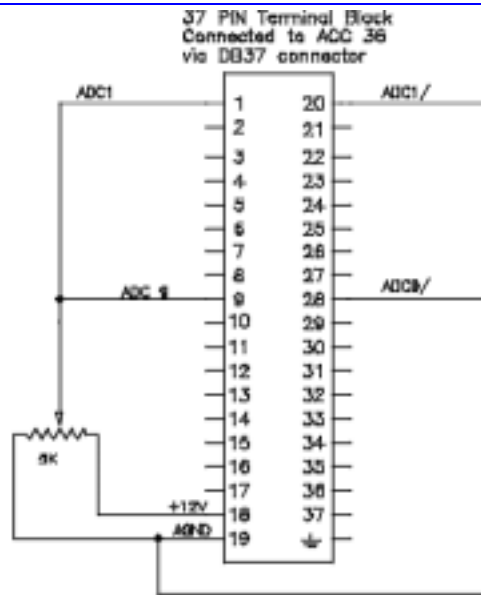
```
WY:$072A,$300708,$000FFF,$000010
```

After the encoder conversion table is set up properly, the following I-Variables must be changed for A/D position feedback.

Variable	Value	Units	Definition
I60	\$FFD0	PMAC Y Addresses	Auto Converted ADC Register Address Range
I61	0	Number of registers minus 1	Number of Auto-Converted ADC Registers
Ix03	\$0720	Legal PMAC X addresses	Position Loop Feedback Address
Ix04	\$0720	Legal PMAC X addresses	Velocity Loop Feedback Address
Ix05	\$072C	Legal PMAC X addresses	Master Handwheel Position Address
Ix06	1	None	Master Handwheel Following Enable
Ix60	85	360/256 elec deg (1/256 commutation cycle)	Motor Servo Cycle Period Extension

As the potentiometer is turned, the motor should move accordingly.

## Wiring Diagram for Acc-36 Servo Feedback Use



Servo Feedback Loop Test Setup for Accessory 36

## Example using Acc 36 for both Data Acquisition and Servo Feedback

For this example, use the first eight channels for servo feedback and the last eight channels for data acquisition. To accomplish this, download the following definitions to PMAC:

### Variables Written to the Encoder Conversion Table

```

WY:$0720,$300708,$000FFF,$000010
WY:$0723,$300709,$000FFF,$000010
WY:$0726,$300708,$000FFF,$000010
WY:$0729,$300708,$000FFF,$000010
WY:$072C,$300708,$000FFF,$000010
WY:$072F,$300708,$000FFF,$000010
WY:$0732,$300708,$000FFF,$000010
WY:$0735,$300708,$000FFF,$000010

```

## M-Variables Used for Data Acquisition

```

m69->x:$708,0,12,u           ; channel 9A to D
m70->x:$709,0,12,u           ; channel 10A to D
m71->x:$70A,0,12,u           ; channel 11A to D
m72->x:$70B,0,12,u           ; channel 12A to D
m73->x:$70C,0,12,u           ; channel 13A to D
m74->x:$70D,0,12,u           ; channel 14A to D
m75->x:$70E,0,12,u           ; channel 15A to D
m76->x:$70F,0,12,u           ; channel 16A to D

```

### Note:

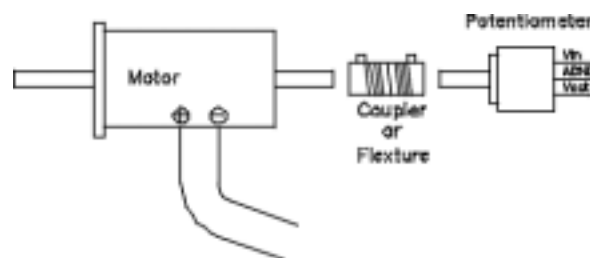
Since the data is being processed as a filtered parallel word, it uses three rows in the encoder conversion table.

**Example:** If encoder channel #1 is sourced from location Y:\$720, the processed encoder information will be at location X:\$722. Therefore, to use a filtered parallel word as feedback, set Ix03 = \$722 and Ix04 = \$722.

Position and Velocity Feedback Variables	Encoder Conversion Table Entry
I103, I104	\$722
I203, I204	\$725
I303, I304	\$728
I403, I404	\$72B
I503, I504	\$72E
I603, I604	\$731
I703, I704	\$734
I803, I804	\$737

## Example Using Acc-36 for True Motor Position Feedback

This example will show how to use this board for motor position feedback. There are many analog devices or sensors which may be used for position feedback. For this example, a ten-turn potentiometer coupled to a motor shaft will be used as shown below.



- Do not couple motor to potentiometer at this point.
- Make sure motor is not connected to load.

First, write to the encoder conversion table. Notice the default end of the encoder conversion table location is being used. Feel free to use an encoder conversion table entry which does not interfere the existing system setup.

```

WY:$0728,$300708,$000FFF,$000010 ;end of default conversion table

```

After adding an entry to the encoder conversion table, now change the position and velocity feedback I-Variables (Ix03 and Ix04) to PMAC to \$0728. A change of the position can be seen in the position window of the executive by turning the potentiometer by hand.

Before connecting the motor shaft to the potentiometer, make sure a positive voltage direction corresponds to a positive motor direction. Do this by issuing open loop commands to the motor in the positive and negative directions. If the motor is spinning in the wrong direction relative to the potentiometer, switch the motor leads going to the motor.

Once the motor direction matches the potentiometer direction, now couple the motor to the sensor and use the sensor for position loop feedback.

## Absolute Power on Position

If absolute power-on position information is desired and usually it is, Ix10 (new in V1.14 firmware) must be set to point to the proper register and to tell PMAC how to interpret that register. If Ix10 is set to 0, PMAC will always treat the power-on position as zero. Ix10 should point to the register containing the data before processing through the conversion table.

In the example, I110 would be set to \$8C0708, where the 8 specifies a signed value from a Y register, the C specifies a 12-bit value, and the \$0708 specifies the address. I210 would be set to \$CC0708, where the first C (8+4) specifies a signed value from an X register, the second C specifies a 12-bit value and the \$0708 specifies the address.

Remember that I-Variables must be stored to non-volatile memory with the **SAVE** command in order to be retained through a power-down cycle or reset.

## Using Acc-36 for Servo Loop Feedback using PLC0

If using 12-bit (0-4096) resolution analog feedback, the ADC registers may be read using PLC0 to achieve this resolution. The method of reading the analog input is the same as described using the regular background PLC to read the ADC registers as they are addressed by the program. The advantage of using this method is the PLC can be scanned every real time interrupt. If a compiled PLCC is used the scan time to read the ADC registers is increased.

Closing the servo loop with Acc-36 using a PLC0 is a more complicated process than the automatic method as described previously. Using this method, feedback resolution will be 12 bits (instead of signed 11 bits using the automatic method). This method involves the use of array writing to defined M-Variables to efficiently read the addressed channel of Acc-36. See the Array Capabilities section of the User manual for a description of this feature.

In addition, the use of I60 applies when reading analog inputs in this fashion. If all eight registers are being read, set I60 equal to 7.

M99->Y:\$FFD0,24	;24 bit register pointing to PMAC's processor bus address
M0->Y:\$07F0,24	;24 bit register equal to the addressed channel
m10->Y:\$BC00,0,4	;array write variable for M0
M1->Y:\$07F0,0,12	;12 bit register for used for feedback (lower 12 bits Y register)
M11->Y:\$BC01,0,4	;array write variable for M1
M2->Y:\$07F0,12,12	;upper 12 bits which will need to be changed to lower 12 bits of the
	;X register (next M-variable)for the ; encoder conversion table.
M12->Y:\$BC02,0,4	;array write variable for M2
M3->X:\$07F0,0,12	;lower 12 bits of X register to be used for encoder conversion

```

M13->Y:$BC03,0,4           ;array write variable for M3
OPEN PLC 0 CLEAR
M0=M99                       ;Set m0 equal to Acc-36's addressed 24-bit register
M3=M2                         ;Set lower 12-bits equal to upper 12-bits of PMAC's addressed channel
M10=(M10+1)&7                 ;Ensure we only increment from 0 to 7
M11=M10                       ;increment array
M12=M10                       ;increment array
M13=M10                       ;increment array
M99=M10                       ;address next channel
CLOSE

```

To use these registers for servo feedback, make an entry to the encoder conversion table. The entry to the encoder conversion table is the either an X or Y parallel word with or without filtering. For this example, assume that AQUADB encoder feedback is being used for the first four channels and wants to use Acc-36 channels 1, 2, 9, and 10 for closed loop feed back.

```

WY:$724,$3007F0,$000FFF,$000010 ; channel 1
WY:$727,$3007F1,$000FFF,$000010 ; channel 2
WY:$72A,$7007F0,$000FFF,$000010 ; channel 9
WY:$72D,$7007F1,$000FFF,$000010 ; channel 10

```

Now use the processed encoder conversion registers for feedback. Since these conversions use three rows each in the encoder conversion table, the feedback (position, velocity, or master) will be located at:

```

$726    for channel 1
$729    for channel 2
$72C    for channel 9
$72F    for channel 10

```

### Part Number for Fuse on Acc-36

Manufacturer	Part Number
Little Fuse	273.500

Specs: 125V @ 0.5A